Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2004 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2004

A Secure Key Management Scheme for Sensor Networks

John Wynne
Boston University

Kristie Kosaka
Claremont Graduate University

Samir Chatterjee
Claremont Graduate University

A Secure Key Management Scheme for Sensor Networks

Alan Price
Claremont Graduate University
alan.price@cgu.edu

Kristie Kosaka
Claremont Graduate University
kristie.kosaka@cgu.edu

Samir Chatterjee
Claremont Graduate University
samir.chatterjee@cgu.edu

ABSTRACT

Key technological advances in wireless communications, Micro Electro Mechanical Systems (MEMS), and digital circuitry have energized the research community to focus on the challenges of wireless sensor networks. In this paper, we propose a new pre-distribution key management scheme that meets the operational and security requirements of wireless sensor networks and provide authentication and key distribution in one set of protocols. Our scheme allows selective key revocation and node re-keying and posits improved network resiliency over existing key pre-distribution schemes. The scheme is based on probability key sharing among sensor nodes of a random graph and incorporates a threshold property. Uncompromised nodes in a sensor network are secure provided that an adversary compromises less than a threshold-number of nodes. We describe the details of our algorithm and briefly compare it with other proposed schemes.

Keywords

Security algorithms, wireless sensor networks, pre-distribution key management

INTRODUCTION

Wireless sensor networks have emerged as an innovative class of networked embedded systems due to the union of ever smaller, less costly embedded processors and wireless interfaces with micro-sensors based on micro-mechanical systems (MEMS) technology (Peters, Smith, Medeiros, and Rohrer, 2001). Wireless sensor networks are composed of small autonomous devices, or sensor nodes, that are networked together. Each node is equipped with one or more sensors, storage and processing resources, and communication and instrumentation subsystems. The sensors observe phenomena; each sensor is specialized to monitor a specific environmental parameter such as thermal, optic, acoustic, seismic, or acceleration (Meguerdichian, Koushanfar, Potkonjak, and Srivastava, 2001). Sensor nodes typically perform their tasks unattended, often in remote locations. They may be deployed either inside, or nearby, target phenomenon to be studied.

Typical sensor networks will support a variety of military, medical, environmental, and commercial applications. Remote sensors could reduce confusion within combat zones by collecting information about battlefield conditions. Sensor networks are currently being used for condition-based maintenance of complex equipment in factories. Natural environments (i.e., remote ecosystems, endangered species, disaster sites, forest fires.) can be monitored with sensor networks (Kahn, Katz and Pitzer, 1999; Park, Savvides, and Srivastava, 2001).

Sensor networks often contain one or more sinks that provide centralized control. A sink typically serves as the access point for the user or as a gateway to another network (Akyildiz, Su, Sankarasubramaniam, and Cayirci, 2003). Large sensor networks can be composed of thousands of sensor nodes deployed in the field to jointly observe a region. Figure 1 depicts a typical sensor network. Compromise of any node,

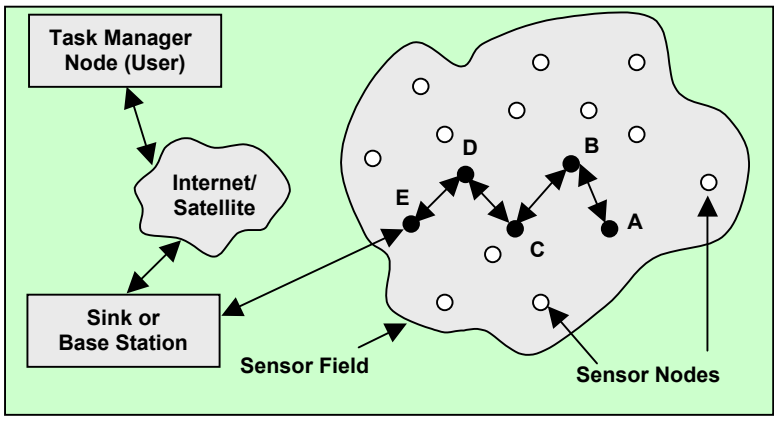

Figure 1 – Sensor Network Architecture

particularly the sink, is a sensor network security concern, but outside the scope of this paper.

Sensor networks have several major constraints awaiting research resolution: limited processing power, limited storage capacity, limited bandwidth, and limited energy. Researchers are working to solve many of the limitations affecting sensor nodes and networks. Some researchers are working to improve node design; others are developing improved protocols associated with a sensor network; still others are working to resolve security issues. The focus of our research involves secure key distribution schemes within wireless sensor networks.

KEY DISTRIBUTION SCHEMES IN SENSOR NETWORKS

Sensor networks are typically wireless deployments, sometimes in hostile environments, and are subjected to greater security risks. Establishing secure communications involving the setup and distribution of secret keys is an open problem for sensor network researchers. Currently, there are three general key agreement schemes: trusted-server or arbitrated protocol, self-enforcing, and key pre-distribution scheme (Du, Ding, Han, and Varshney, 2003). The trusted-server scheme requires a trusted server to establish shared-session keys between nodes and is prone to directed attacks against this central point of weakness. Another key agreement scheme is the self-enforcing scheme, which depends upon asymmetric protocols and algorithms. However, with the low memory and energy constraints of sensor nodes, public-key algorithms common in asymmetric cryptography limits the practical use of this key distribution scheme. Presently, the only practical scheme for key distribution in large sensor networks is key pre-distribution, where key information is installed in each sensor node prior to deployment. Typically, two solutions have been used: 1) a single mission key where all nodes carry a master secret key or 2) a set of separate n – 1 keys, each being a pairwise set that is privately shared with another sensor node (Eschenauer and Gilgor, 2002; Du, Fang, Wang, and Chen, 2003). Both are inadequate for use in sensor networks since conciliation of the single mission key may compromise the entire network and storage of n – 1 keys in each sensor node or n(n – 1)/2 per sensor network bounds practical adoption. To overcome the challenges and limitations associated with both schemes, several other key management schemes have been proposed.

Eschenauer and Gilgor's Random Key Pre-Distribution Scheme

Eschenauer and Gilgor (Eschenauer and Gilgor, 2002) proposed a random key pre-distribution scheme based on probabilistic key sharing and utilization of a simple shared-key discovery protocol for key distribution, key revocation, and node re-keying. Prior to a sensor network deployment, each sensor node receives a key ring with a randomly chosen subset of keys from a large key pool. Upon deployment and network initialization, sensor nodes will be able to establish a secure and direct communication link provided that a shared key exists between one or more pairs of sensor nodes. Due to the random distribution of keys to each sensor node, it is probable that a shared key may not be available, necessitating an intermediary node with a common key between the two sensor nodes to establish a common session key. Eschenauer and Gilgor found that to establish "almost certain shared-key connectivity for a 10,000-node network, a key ring of only 250 keys randomly selected from a 100,000 pool has to be pre-distributed to every sensor node."

Eschenauer and Gilgor's key distribution scheme consists of three phases: key pre-distribution, shared key-discovery, and path-key establishment. The *key pre-distribution phase* occurs prior to sensor node deployment. During this phase, a large pool of *P* random keys (e.g., $2^{17} - 2^{20}$ keys) and their key identifiers are generated. Each sensor node receives a subset of randomly chosen *k* keys and their associated key identifiers plus a shared key with a trusted controller node that stores all keys and associated identifiers for every network node. The shared key between the controller node and each sensor node is unique and is infrequently used to support key revocation. The purpose of the key pre-distribution phase is to ensure that a small number of keys are available to probabilistically establish a common key between two or more sensor nodes during the shared-key discovery phase.

The *shared-key discovery* phase occurs post-hoc of the sensor network deployment during an initialization period where each sensor node attempts to discover its neighbors with which its shares a common key(s). This can be done by the broadcast of each sensor nodes key identifier list in plaintext or a list α, E_{Ki} (α), i = 1, … , k, where α is a challenge. Decryption with a proper key of E_{Ki} (α) would meet the challenge and establish a shared key with the broadcasting node. It is possible that more than one pair of sensor nodes may share the same key since all keys are randomly chosen from a larger pool set.

The *path-key establishment* phase is used to assign a path-key to selected sensor-node pairs within a defined communication range that do not share a common key but are connected by two or more links created during the shared-key discovery phase. An intermediary node generates the path-key with a shared key between two or more unconnected link nodes. Path-keys do not have to be generated by the intermediary, as a number of keys are available on its key ring after the shared-key discovery phase is finished.

Blom's Symmetric Key Generation Scheme

Du et al. (2003) proposed a pre-distribution scheme that adapted ideas from Blom's symmetric key generation system (Blom, 1985) and Eschenauer and Gilgor's algorithm previously discussed. According to Blom, any pair of nodes can calculate a secret pairwise key between them using distinct data elements stored in λ + 1 memory spaces in each node. Blom posited that "as long as no more than λ nodes are compromised, the network is perfectly secure (referred to as the λ-secure property)." Increasing λ leads to greater network resiliency but also leads to higher memory utilization within each sensor node. To calculate a secret pairwise key, each sensor node uses a data set derived from several linear algebra operations.

1. During the pre-deployment phase, a (λ + 1) x N matrix *G* over a finite field GF(q), where q is an element within the finite field, N is the number of sensor nodes in the network and λ is the security parameter previously discussed, is constructed. To meet the λ-secure property, *G* must be linearly independent. It has been shown that a Vandermonde matrix is linearly independent when its elements s, s^2, s^3, … , s^N are all distinct (MacWilliams and Sloane, 1977). A feasible (λ + 1) x N matrix *G* over a finite field GF(q) can be constructed based on this type of matrix (Du et al, 2003; MacWilliams and Sloane, 1977) depicted in Equation 1. Each nonzero element of GF(q) can be represented by some power of s^i for some $0 < i \leq q - 1$, where q is chosen to be the smallest prime number larger than $2^{\text{key size}}$.

$$G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ s & s^2 & s^3 & \cdots & s^N \\ s^2 & (s^2)^2 & (s^3)^2 & \cdots & (s^N)^2 \\ & & & \vdots & \\ s^\lambda & (s^2)^\lambda & (s^3)^\lambda & & (s^N)^\lambda \end{bmatrix}$$

Equation 1

2. During pre-deployment, a random (λ + 1) x (λ + 1) symmetric matrix *D* over GF(q) is computed and used to compute an N x (λ + 1) matrix *A*, which is equal to $A = (D \cdot G)^T$. Matrix *D* is private information and must be kept secret. Matrix *A* is the transpose of $(D \cdot G)$. Therefore, the k^{th} column of $(D \cdot G)$ becomes the k^{th} row of $(D \cdot G)^T$.
3. During pre-deployment, the k^{th} row of matrix *A* and the k^{th} column from matrix *G* is stored at sensor node k, where k = 1, … , N. In practice, s^k is a primitive or seed element of GF(q) that can be used to calculate the associated column values of matrix *G*. Therefore, only s^k needs to be stored at sensor node k.
4. After sensor node deployment, two nodes *i* and *j* can find the pairwise key between them by exchanging their columns of *G* and using their private rows of matrix *A* to calculate $k_{ij} = k_{ji} = Ai \cdot Gj = Aj \cdot Gi = (A \cdot G)^T$. This is possible since $A \cdot G$ is a symmetric matrix. Figure 2 illustrates how secret pairwise keys are generated.

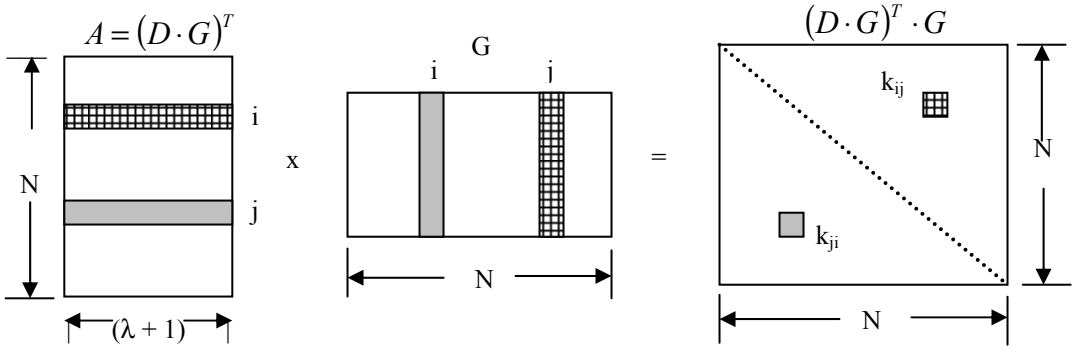

Figure 2

Multiple-Space Key Pre-distribution Scheme

Under Du et al. multiple key spaces generated from Blom's λ-secure symmetric key generation system are randomly assigned to each sensor node in a network. This is similar to Eschenauer and Gligor's assignment of randomly generated keys from a large key pool. Two nodes are able to calculate a unique pairwise key if and only if both nodes share a common

key space. During the pre-deployment phase, a generator matrix G of size (λ + 1) x N is created and followed by the creation of ω symmetric matrices $D_1, \ldots, D_\omega$ of size (λ + 1) x (λ + 1) and matrices $A_i = (D_i \cdot G)^T$, where indices i = 1, … , ω, are generated. Each tuple of A_i is defined as a key space. From A_i, a randomly selected number of $1 < \tau \leq \omega$ distinct key spaces are selected and assigned to each sensor node. After deployment, each node will attempt to identify its common-space neighbors by broadcasting a message that contains its unique node ID, the indices of the key spaces it carries, and the assigned column data set or seed value from G. If nodes i and j are neighbors with a common space (e.g., S_c, with node i assigned $A_c(i)$ and seed for $G(i)$ and node j has $A_c(j)$ and the seed for $G(j)$), a secret pairwise key can be calculated by each node independently by $k_{ij} = k_{ji} = A_c(i) \cdot G(j) = A_c(j) \cdot G(i)$.

OUR PROPOSED KEY PRE-DISTRIBUTION SCHEME

We propose a new key pre-distribution scheme that builds from Blom's and Eschenauer and Gilgor's key distribution schemes and is supported by points based on key splitting, authentication and key exchange from the Yahalom protocol (Schneir, 1996). Our scheme leverages Eschenauer and Gilgor's reliance on probabilistic key sharing between nodes that have been assigned a randomly selected subset of keys from a larger key pool but, unlike Eschenauer and Gilgor's scheme of storing k, 64-bit keys, we store 32-bit keys in each node and concatenate an additional 32-bit random key for links that are able to establish a secure communication link. A unique 64-bit pairwise key that contains the shared key-half between nodes and a second key-half generated after deployment and during a shared-key discovery phase supports each secure link. The second key-half is randomly generated by one of the node-pairs and distributed with support from Blom's key calculation scheme. Following is the detail of our key pre-distribution proposal in relationship to the three basic phases previously discussed: key pre-distribution phase, shared-key identification phase, and path-key establishment phase. The path-key establishment phase will not be discussed as it assumes the same protocol proposed by Eschenauer and Gilgor.

1. **Generation of Large Key Pool.** A large pool of random 32-bit keys denoted by P and associated identifiers are generated. The size of the pool is sufficiently large (e.g., 100 – 1,000 times the deployed network size) to ensure non-probability key attacks are minimized. Each key will be identified by an integer value 1 to the number of keys generated.

2. **Generation of Blom's System Matrices.** A primitive element, s, is selected from a finite field GF(q) such that the element is the smallest prime number larger than 2^{32}. This element represents the seed element of the general Vandermonde matrix G shown in Equation 1. The size of the matrix will equal the number of P keys generated in step 1 and an λ equal to a security threshold level appropriate for the network (e.g., $\lambda \geq 250$). Once a seed value has been selected, a random (λ + 1) x (λ + 1) symmetric matrix D is generated and used to compute an N x (λ + 1) matrix $A = (D \cdot G)^T$. All elements in each matrix are considered non-public information and must be kept secret.

3. **Node Assignments.** For each sensor node, k randomly selected keys and associated identifiers are assigned. These keys represent one-half of the pairwise keys that will be generated between two nodes. Each node will be assigned one row from matrix A corresponding to one key and associated key identifier generated in step 1. This row-key pair will be used as a primary identifier for each sensor node. Lastly, the primitive element selected in step 2 will be stored in each sensor node's memory. This element will be used to calculate the corresponding column of matrix G for each key identifier selected. A general memory map for each sensor node is shown in Figure 3, where A_x is the row assignment from matrix A and ID_x, and key (x) are k randomly selected keys and associated integer identifiers over $1 < x < k$.

32 bits	32 bits
Primary Key A_{x1}, Id_{x1}, key (x_1)	Randomly generated key y_1
Id_{x2}, key (x_2)	Randomly generated key y_2
.	.
.	.
.	.
Id_{xk}, key (x_k)	Randomly generated key y_K

Figure 3

Shared-Key Identification Phase

After sensor network deployment , each node needs to establish a secure communication link with each neighbor who share a common key and are within wireless communication range. This is started with each node broadcasting a message containing their primary key identifier and a randomly generated nonce value N_x. Assuming that nodes i and j are neighbors, and have received the broadcast, they will check their memory map for a common key using the received key identifier. If they find that they share a common key, they will each calculate a secret pairwise secret key using Blom's scheme. This

secret key will be concatenated (symbolized below by ||) with the shared common key and used to encrypt a message that contains a new nonce value associated with the responding node, an identification value used to uniquely identify the communication link between the nodes, and a random session key S_k that will be used as the second-half of the secret key between each node. The first half of the key will be the common key shared between both node-pairs. Assuming that Node 1 and Node 2 each share a common half-key, the following details the steps that both nodes will take in establishing a secure communication link.

1.) Node 1 broadcast $ID_1||N_1$, where N_1 is a nonce value generated by Node 1 and ID_1 is equal to integer 1 and is the identifier assigned to Node 1 and its primary key.

2.) Node 2 picks up $ID_1||N_1$, sees it has a key associated with ID_1 and sends back to Node 1: $ID_2||N_1||E_{12}(ID_{12}, N_2, S_k)$, where ID_{12} is a random number used as a link identifier between both nodes, N_2 is a nonce value generated by Node 2, S_k is a random 32-bit session key, and $E_{12}( )$ is a symmetric encryption function. The encryption key for $E_{12}( )$ is calculated as follows:

   a. Using the Vandermonde matrix in Equation 1, Node 2 calculates the matrix *G* column associated with Node 1 using the primitive element *s* stored in Node 2 by the primary integer identifier sent from Node 1 and raising this seed element to power *n* for $2 < n \leq \lambda$.

   b. Node 2 calculates a pairwise secret key between Node 1 and Node 2 using $K_{12} = K_{21} = A(2) \cdot G(1)$, where A(2) is Node 2 assigned row from matrix *A* and G(1) is Node 1's column data.

   c. Node 2 generates an encryption key for $E_{12}( )$ by concatenating the shared key between both nodes and the calculated K_{12} key.

3.) Node 1, having received $ID_2||N_1||E_{12}(ID_{12}, N_2, S_k)$ and using the integer identifier associated with Node 2, calculates the key for $E_{12}( )$ following steps 2a-2c. Node 1 decrypts $E_{12}( )$ to retrieve ID_{12}, N_2, and S_k.

4.) Node 1 sends back an acknowledgement message to Node 2 of $N_2||E_k(ID_{12}, N_1)$. This completes the authentication and key exchange between both nodes.

The authentication and key exchange listed above assumes that a shared key will be discovered between two nodes such that the shared key is a primary key for one node and an ancillary key for the second node. In the event that nodes within communication range cannot establish a primary to secondary key agreement, nodes will attempt to establish a secondary-to-secondary key agreement by broadcasting for each ancillary key in its key set a message containing $ID_X||N_X||ID_{Primary}$, where ID_X is a ancillary key identifier, N_X is a nonce value associated with ID_X, and $ID_{Primary}$ is a nodes primary identification number. Once two nodes determine that they share the same key-half, steps 1 – 4 will be followed with $ID_X||N_X||ID_{Primary}$ replacing $ID_1||N_1$ in step 1. The primary node identifier is used for all Blom matrix calculations.

Node Revocation

In the event that a sensor node is compromised, it is important to revoke that hostile node. Our scheme supports two approaches to effect node revocation. Link revocation removes all communication links, while key revocation removes all key(s) associated with the compromised node. Link revocation allows only selected node-pair links to be removed thereby minimizing the impact of revocation on the network. This can be accomplished since each node-pair shares a mutually exclusive link identifier and session key. While multiple node-pairs may share the same half-key, their other key-half will be distinct to each node-pair. Key revocation, in comparison, removes all shared half-key(s) associated with the compromised node, which may cause multiple links to disappear once the key(s) are removed. Both methods assume that a secure base station is used to monitor and manage the sensor network.

Link revocation can be implemented using a network routing table generated after post-deployment of the sensor network and link establishment between node-pairs. The base station creates a network routing table by having each node-pair identify their associated link identifier and having this information sent over the network to a base station. Key revocation follows the same procedure but one or more keys and not associated links are removed during the revocation process. Once the keys have been removed from the memory space of each node, some links may disappear and will require the affected nodes to establish new links by restarting the shared-key discovery and/or path-key establishment phases.

Several options are available to revoke the compromised node including the use of a shared signature key, which is installed between a base station and all nodes of the sensor network during the pre-deployment phase. To initiate revocation of a link or key(s), the base station will broadcast to all network nodes a signed revocation message containing the link identifier(s) or key(s) to be revoked. After obtaining the signed revocation message, each node will verify the signature of the signed

message and, if valid, will remove the link identifier(s) or key(s) from their memory. This method is simple to implement but poses a security risk due to its reliance on a single signature key. The second option uses a signed unicast message that contains the link identifiers(s) and associated session key(s) that need to be revoked between a base station and each affected sensor node in the network. Each sensor node will share a unique key with the base station that can be installed during the pre-deployment phase or computed as $K_{shared} = E_{Kx}$ (ci), where $Kx = K_1 \oplus, \ldots, \oplus K_k$, K_i are the keys assigned to the sensor node, ci is the base station identifier, and E_{Kx} is an encryption function with node key K_x. Security is increase with method two and was proposed and used by Eschenauer and Gilgor in their key management scheme.

SECURITY ANALYSIS OF OUR KEY PRE-DISTRIBUTION SCHEME

Our key pre-distribution scheme, at its fundamental level, is a hybrid between Eschenauer and Gilgor's promulgation of random key distribution and Blom's generation of pairwise secret keys. Under Eschenauer and Gilgor's scheme, a node carries a subset of *k* randomly sampled keys from a larger pool of *P* keys that, if compromised, will allow an adversary to attack successfully with probability of $\frac{k}{P}$ any sensor network link. Since multiple nodes may share one or more keys, compromising one node may allow an adversary to attack $\frac{k \cdot shared\ number\ of\ links}{p}$ links. Their analysis suggests, "compromise of one key does lead to the compromise of another link with probability 0.3, of two other links with probability 0.1, and so on." Under Blom's key generation scheme, all pairwise keys can be obtained if more than λ nodes are compromised. To improve security, λ can be increased but this requires increased memory space. Our scheme achieves better resilience against node capture over existing key pre-distribution schemes since compromise of one node yields only *k* keys out of a larger pool of *P* keys, which yield little value without additional compromise of more than λ nodes. One attack potential not addressed by Blom is the compromise of one node and its associated row information of matrix *A* may allow an adversary to masquerade as the captured node and to establish a secure communication link with any sensor node in the network. Undetected, the masquerading node could initiate passive and active attacks including traffic analysis, modification of messages, or general disruptions of communications (Schneir, 1996). Our scheme reduces the probability of this type of an attack; compromising the k^{th} row information of matrix *A* at node *k* is not sufficient to establish a communication link with another sensor node. An adversary would need to compromise the captured sensor nodes random key set and find another node in the network with a common key or key set to integrate a captured node into the network.

Regardless of the scheme, additional weaknesses are seen with the amount of memory utilization required to store *k* random keys or λ + 1 finite elements from row information of matrix *A*. Using 250 as the benchmark for the number of keys or λ + 1 finite elements stored in each sensor node, and assuming each key and each finite element in λ + 1 is 64-bits, 2 kilobytes of RAM would be required. While a security analysis has not been done to determine what λ-secure value is needed to equal the same security level for a defined quantity of keys under Eschenauer and Gilgor's scheme, it is predicted that a large λ-value would be needed to meet the same security level between both schemes. Due to the limited memory capacity associated with sensor nodes, increasing the λ-value may not be possible.

Our proposal uses substantially less memory space. Under our scheme, each sensor node receives a randomly sampled subset of 32-bit keys, compared to 64-bits under the Eschenauer and Gilgor scheme. Once a secure link is established, an additional 32-bits is added to create a 64-bit session key but the number of established session keys that must be stored in each node is much smaller than storing a *k* set of 64-bit keys, many of which will never be used once the network field has been established. While our scheme requires the storage of λ + 1 finite elements, it is hypothesized that a small λ-value can be chosen to meet a comparable security threshold level.

BENEFITS SUMMARY OF OUR KEY PRE-DISTRIBUTION SCHEME

This paper posited the details of a new key pre-distribution scheme with the following properties:

1. Authentication and key distribution in one set of protocols. Only nodes that share a common key, or use an intermediary with a common key, will be able to establish a secure channel.
2. Establishment of unique communication links with different key-sets that can support future network routing establishment and study. Each communication link between each node-pair is uniquely identified and secure with a session key composed of a first-half key common to both and a second half that is randomly generated. Key distribution relies on a nodes ability to calculate a temporary cryptographic key from a set of linear algebra

operations defined in Blom's key generation scheme and is supported by the λ-secure property. The link identifiers can be used for message routing or link revocation in the event of a compromised node-pair.

3. Theoretically improved network resiliency and lower memory utilization over existing key pre-distribution schemes. Our scheme achieves better resilience against node compromise over Eschenauer and Gilgor's pre-distribution scheme since compromise of one node yields only *k* keys out of a larger pool of *P* keys. This provides little value without the additional compromise of more than λ nodes, as specified by Blom's scheme and supported in our algorithm. Lastly, each sensor node receives a randomly sampled subset of 32-bit keys, compared to 64-bits under the Eschenauer and Gilgor scheme. Once a secure link is established, an additional 32-bits is added to create a 64-bit session key but the number of established session keys that must be stored in each node is much smaller than storing a *k* set of 64-bit keys, many of which will never be used once the network field has been established.

CONCLUSION

Nodes within wireless communication range will be able to establish a shared-key connectivity provided that they meet the points listed above. While limitations of a sensor network (e.g., wireless communication range, mutually exclusive key sets, etc.) may preclude a network with 100% shared-connectivity, it should be possible to calculate the required key size, *k*, pool size, *P*, and λ-secure level for a sensor node with memory size, *m*, so that *k* nodes are connected. Eschenauer and Gilgor showed using random graph theory that share-connectivity was "almost certain" for a 10,000-node network with a key ring of 250 keys drawn out of a pool of 100,000. While it is expected that the conclusions proposed by Eschenauer and Gilgor would also hold in our proposed scheme, a detailed mathematical study is required for validation. We are presently developing mathematical models that will analyze our algorithm and simulation models to test its performance. Additional work is required to validate the premise that our proposed scheme provides greater resiliency to security attacks than other schemes addressed in this paper.

REFERENCES

1. Peters, B., Smith, J., Medeiros, O. and Rohrer, M. (2001) Improving Simulation Model Adaptability With a Production Control Framework, *Proceedings of the 2001 Winter Simulation Conference*, http://www.isr.umd.edu/~jwh2/papers/WSC2001.pdf.

2. Meguerdichian, S., Koushanfar, F., Potkonjak, M. and Srivastava, M.I. (2001) Coverage Problems in Wireless Ad-Hoc Sensor Networks, IEEE Infocom 2001, Vol 3, 1380-1387.

3. Akyildiz, I., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2003) A survey on Sensor Networks, *IEEE Communications Magazine*, 102 – 114.

4. Kahn, J. M., Katz, R. H. and Pister, K. S. J. (1999) Mobile Networking for Smart Dust, *ACM/IEEE International Conference on Mobile Computing (MobiComm '99),* Seattle, WA, 217 – 278.

5. Park, S., Savvides, S. and Srivastava, M. (2001) Simulating networks of wireless sensors, *Proceedings of the 2001 Winter Simulation Conference*, 1330-1338.

6. Du, W., Ding, J., Han, Y. and Varshney, P. (2003) A pairwise key pre-distribution scheme for wireless sensor networks, *CCS '03*, Washington, DC.

7. Schneir, B. (1996) Applied Cryptography Second Edition: protocols, algorithms, and source code in C, John Wiley & Sons, Inc.

8. Eschenauer, L. and Gilgor, V.D. (2002) A key-management scheme for distributed sensor networks, *Proceedings of the ACM conference on Computer and communication security.*

9. Du, W., Fang, L., Wang, R. and Chen, S., Key pre-distribution using sensor pre-deployment knowledge. Downloaded from: web.syr.edu/~rwang01/Research/WiSe03.pdf, October, 2003.

10. Blom, R. (1985) An optimal class of symmetric key generation systems, *Advances in Cryptography: Proceedings EUROCRYPT 84 (Thomas Beth, Norbert Cot, and Ingemar Ingemarsson, eds.), Lecture Notes in Computer Science, Springer-Verlag*, 209:335-338.

11. MacWilliams, F.J. and Sloane, N.J.A. (1977) The Theory of Error-Correcting Codes, *Elsevier Science Publishing Company, Inc*, New York, NY.